IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.	: 10/660,289	Confirmation No.:	5354
Applicant	: Brent Russell Phillips		
Filed	: September 11, 2003		
Title	: METHODS, SYSTEMS, AND MEDIA TO ENHANCE PERSISTENCE OF A MESSAGE		
Art Unit	: 2444		
Examiner	: Bayard, Djenane M.		
Docket No.	: AUS920030563US1(4013)		
Customer No.	: 45557		

Mail Stop Appeal Brief - Patents
Commission for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. §41.37

This paper is submitted pursuant to 37 CFR §41.37 in furtherance of the Notice of Appeal filed on April 9, 2009, for the above referenced patent application to appeal final rejections imposed by the USPTO on claims in the above referenced patent application to the Board of Patent Appeals and Interferences ("Board") after careful consideration to address issues associated with the final rejections. In light of the timely submission of this Appeal Brief pursuant to 37 CFR § 41.37 and the arguments contained herein, Appellant respectfully requests that the Board reverse the rejections of the pending claims and remand this application to the Examiner for reconsideration consistent with an order that the Examiner pass this case to issuance unless a proper rejection to the claims can be made.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation ("IBM") having a principle place of business at New Orchard Road, Armonk, NY 10504, as assignee of patent(s) resulting from the above-referenced patent application, in view of the assignment executed by the inventor to IBM.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals nor interferences known to Appellants, Appellants' legal representative, or assignee which will directly affect or be directly affected by or having a bearing on the Board's decision in this pending appeal.

III. STATUS OF CLAIMS

Claims 1-20 are pending and claims 1-20 stand rejected. Claims 1-20 are appealed herein. Claims 1-20 stand rejected by a final Office action dated November 26, 2008. More particularly:

1) Claims 1-6, 9-15, and 17-20 stand rejected under 35 USC § 103(a) as being unpatentable over McGann et al., U.S. Pat. 6,920,476 (hereinafter "McGann") in view of Lambert et al., U.S. Pat. Pub. 2003/0033349 (hereinafter "Lambert").
2) Claim 7 stands rejected under 35 USC § 103(a) as being unpatentable over McGann in view of Lambert and in further view of Nakada, U.S. Pat. Pub. 2001/0013051 (hereinafter "Nakada").
3) Claim 16 stands rejected under 35 USC § 103(a) as being unpatentable over McGann in view of Lambert and in further view of Mikalsen, U.S. Pat. 6,934,948 (hereinafter "Mikalsen").

IV. STATUS OF AMENDMENTS

All amendments filed up through the final Office action dated November 26, 2008 have been entered. No after final amendment was filed. No other amendments have

been filed subsequent to the final rejection. The claims found in the Exhibit of this Appeal Brief reflect the appealed claims as they are understood by the Appellants at the date of this appeal.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Appellants' independent claim 1 as currently presented claims a method for enhancing persistence of a message. (*See, e.g.,* Specification, pp. 4-5, par. 12, first sent.).[1] The method involves storing the message in an inbound queue after receiving the message. (*See, e.g.,* Specification, pg. 6, par. 16). The method involves browsing the inbound queue to identify the message after storing the message in the inbound queue. (*See, e.g.,* Specification, pg. 7, par. 19, second sent.). The method involves copying the message to a working queue (*See, e.g.,* Specification, pg. 7, par. 18, third sent.), the working queue being persisted by a queue manager (*See, e.g.,* Specification, pg. 7, par. 17), to persist the message, the message being stored, in it's entirety, in both the inbound queue and the working queue concurrently. (*See, e.g.,* Specification, pp. 7-8, par. 20, first sent., and par. 31). The method involves removing the message from the inbound queue after copying the message to the working queue. (*See, e.g.,* Specification, pp. 7-8, par. 20, first sent.). The method involves processing the message to generate a reply prior to removing the message from the working queue. (*See, e.g.,* Specification, pg. 8, par. 21, first three sent.). And the method involves storing the reply in an outbound queue after generating the reply. (*See, e.g.,* Specification, pg. 8, par. 21, first three sent.). Furthermore, as described in claim 2, a further embodiment involves removing the message from the working queue after storing the reply in the outbound queue. (*See, e.g.,* Specification, pg. 8, par. 21, first three sent.).

Appellants' dependent claim 3 incorporates the claims limitations of claim 1 and adds restoring the message in the working queue after a system failure. (*See, e.g.,* Specification, pg. 8, par. 21). Appellants' dependent claim 4 incorporates the claims limitations of claim 1 and adds determining that the message is persisted prior to removing the message from the inbound queue. (*See, e.g.,* Specification, pg. 11, par. 31). Appellants' dependent claim 5 incorporates the claims limitations of claim 1 and adds

[1] Note that "Specification" hereinafter refers to Application at issue, i.e., Application no. 10/660,289 filed September 11, 2003.

wherein browsing comprises searching the working queue for the message as part of a wave of messages in a chronologically adjacent order to facilitate generation of the reply, wherein the message is waiting to be processed. (*See, e.g.,* Specification, pp. 10-11, par. 30). Appellants' dependent claim 6 incorporates the claims limitations of claim 1 and adds locking the message until the message is copied to the working queue. (*See, e.g.,* Specification, pp. 10-11, pars. 30-31). Appellants' dependent claim 7 incorporates the claims limitations of claim 1 and adds wherein processing comprises assigning the message to a thread, the thread being available to process the message. (*See, e.g.,* Specification, pg. 8, par. 22). And, Appellants' dependent claim 8 incorporates the claims limitations of claim 1 and adds wherein processing comprises transmitting a second message to request data indicated by a content of the message and generating the reply based upon data received in response to the second message. (*See, e.g.,* Specification, pp. 13-14, par. 39).

Appellants' independent claim 9 as currently presented claims an apparatus for enhancing persistence of a message. (*See, e.g.,* Specification, pp. 4-5, par. 12, first sent.). The apparatus comprises an inbound queue to receive the message from a requestor. (*See, e.g.,* Specification, pg. 5, par. 14, sent. 2-5, par. 16, and par. 18, sent. 1-4). The apparatus comprises a working queue to store the message. (*See, e.g.,* Specification, pg. 7, par. 18, third sent.). The apparatus comprises an outbound queue to store a reply responsive to the message. (*See, e.g.,* Specification, pg. 7, par. 18, fourth sent., and pg. 8, par. 21, first three sent.). The apparatus comprises a queue manager to persist the message from the inbound queue and the working queue before the message is removed from the inbound queue. (*See, e.g.,* Specification, pp. 7-8, pars. 17 and 20). The apparatus comprises a dispatcher to browse the inbound queue to identify the message after the message is stored in the inbound queue (*See, e.g.,* Specification, pg. 7, par. 19, second sent.); copy the message to the working queue (*See, e.g.,* Specification, pg. 7, par. 18, third sent., and par. 19, first sent.) to persist the message, the message to be stored, in it's entirety, in both the inbound queue and the working queue concurrently (*See, e.g.,* Specification, pp. 7-8, par. 20, first sent., and par. 31); remove the message from the inbound queue after the message is copied to the working queue and after the message is persisted from the working queue (*See, e.g.,* Specification, pp. 7-8, par. 20, first sent.); and assign a thread to process the message to generate the reply in response to the message prior to removing the message

from the working queue (*See, e.g.,* Specification, pg. 7, par. 19, and pg. 8, par. 21, first three sent.) and to store the reply in the outbound queue after generating the reply. (*See, e.g.,* Specification, pg. 8, par. 21, first three sent.). Furthermore, as described in claim 10, a further embodiment involves the outbound queue storing the reply until the reply is transmitted to the requestor. (*See, e.g.,* Specification, pg. 7, par. 18, fourth sent.).

Appellants' dependent claim 11 incorporates the claims limitations of claim 10 and adds wherein the queue manager is configured to persist the message from the inbound queue and the reply from the outbound queue. (*See, e.g.,* Specification, pg. 7, pars. 17-18). Appellants' dependent claim 12 incorporates the claims limitations of claim 9 and adds wherein the dispatcher comprises a persistence determiner coupled with the queue manager to determine that the message is persisted prior to removing the message from the inbound queue, wherein the dispatcher is coupled with the working queue to wait until the thread is available to assign to the message or the thread is being cleaned, before copying the message to the working queue. (*See, e.g.,* Specification, pp. 11-12, pars. 31-32). Appellants' dependent claim 13 incorporates the claims limitations of claim 9 and adds wherein the dispatcher comprises a queue searcher to identify the message to be processed based upon priorities associated with messages stored in the inbound queue. (*See, e.g.,* Specification, pp. 10-11, par. 30). Appellants' dependent claim 14 incorporates the claims limitations of claim 9 and adds wherein the dispatcher comprises a message locker to lock the message, until the message is copied into the working queue. (*See, e.g.,* Specification, pp. 10-11, pars. 30-31). Appellants' dependent claim 15 incorporates the claims limitations of claim 9 and adds wherein the dispatcher comprises recovery logic to assign the thread to process the message after a system failure. (*See, e.g.,* Specification, pg. 12, par. 34). And, Appellants' dependent claim 16 incorporates the claims limitations of claim 9 and adds wherein the thread is configured to process the message based upon a rule associated with the message. (*See, e.g.,* Specification, pp. 12-13, par. 36).

Appellants' independent claim 17 as currently presented claims A storage-type machine-accessible medium containing instructions, which when executed by a machine, cause said machine to perform operations, (*See, e.g.,* Specification, pg. 18, pars. 56-57.) comprising storing the message in an inbound queue after receiving the message. (*See, e.g.,* Specification, pg. 6, par. 16.). The operations comprise browsing the inbound queue to identify the message after storing the message in the inbound queue. (*See, e.g.,*

Specification, pg. 7, par. 19, second sent.). The operations comprise copying the message to a working queue (*See, e.g.,* Specification, pg. 7, par. 18, third sent.), the working queue being persisted by a queue manager (*See, e.g.,* Specification, pg. 7, par. 17), to persist the message, the message being stored, in it's entirety, in both the inbound queue and the working queue concurrently. (*See, e.g.,* Specification, pp. 7-8, par. 20, first sent., and par. 31). The operations comprise removing the message from the inbound queue after copying the message to the working queue. (*See, e.g.,* Specification, pp. 7-8, par. 20, first sent.). The operations comprise processing the message to generate a reply prior to removing the message from the working queue. (*See, e.g.,* Specification, pg. 8, par. 21, first three sent.). And the operations comprise storing the reply in an outbound queue after generating the reply. (*See, e.g.,* Specification, pg. 8, par. 21, first three sent.). Furthermore, as described in claim 18, a further embodiment involves removing the message from the working queue after storing the reply in the outbound queue. (*See, e.g.,* Specification, pg. 8, par. 21, first three sent.).

And, Appellants' dependent claim 20 incorporates the claims limitations of claim 17 and adds wherein browsing comprises selecting a set of messages, the message being part of the set. (*See, e.g.,* Specification, pp. 10-11, par. 30).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

4) Claims 1-6, 9-15, and 17-20 stand rejected under 35 USC § 103(a) as being unpatentable over McGann in view of Lambert.
5) Claim 7 stands rejected under 35 USC § 103(a) as being unpatentable over McGann in view of Lambert and in further view of Nakada.
6) Claim 16 stands rejected under 35 USC § 103(a) as being unpatentable over McGann in view of Lambert and in further view of Mikalsen.

VII. ARGUMENT

The Office action rejected 1-6, 9-15, and 17-20 stand rejected under 35 USC § 103 as being unpatentable over McGann in view of Lambert. Applicant traverses the rejections with the arguments above in conjunction with the arguments below.

To establish a prima facie case of obviousness, the modification or combination must teach or suggest all of Applicants' claim limitations.[2]

McGann in view of Lambert does not describe, expressly or inherently, all of the limitations of claims 1, 9, and 17. For instance, with regards to claims 1 and 17, McGann in view of Lambert fails to teach or suggest:

> ...storing the message in an inbound queue after receiving the message;
> browsing the inbound queue to identify the message after storing the message in the inbound queue;
> copying the message to a working queue, the working queue being persisted by a queue manager, to persist the message, the message being stored, in it's entirety, in both the inbound queue and the working queue concurrently;
> removing the message from the inbound queue after copying the message to the working queue;
> processing the message to generate a reply prior to removing the message from the working queue; and
> storing the reply in an outbound queue after generating the reply.

With regards to claim 9, McGann in view of Lambert fails to describe:

> a dispatcher to browse the inbound queue to identify the message after the message is stored in the inbound queue; copy the message to the working queue to persist the message, the message to be stored, in it's entirety, in both the inbound queue and the working queue concurrently; remove the message from the inbound queue after the message is copied to the working queue and after the message is persisted from the working queue; and assign a thread to process the message to generate the reply in response to the message prior to removing the message from the working queue and to store the reply in the outbound queue after generating the reply.

McGann receives a message at a messaging collector and passes the message to a local queue manager (See FIG. 2, elements 26, 28, 30, and 32). "Messaging collector 28 immediately passes the message off to local queue manager 30. Because local queue manager 30 is local to sending process 20, this happens very quickly and results in minimal latencies. Once messaging collector 28 has passed the message off to local queue manager 30, sending process 20 has completed sending message 26, and may

[2] *In re Royka*, 490 F.2d 981, 985, 180 USPQ 580, 583 (CCPA 1974).

return to other functions."[3] "The messaging collector method 28 preferably returns a status code to sending process 20 indicating success or failure for accepting the message. This is based upon a similar success or failure status received from the local queue manager."[4] "Local queue manager 30 accepts messages as quickly as possible from messaging collector 28, and prepares them for continued transmission."[5] "Also, preferably, [the message] is persisted to a local file system 32 [by the local queue manager 30], where it is stored until the message is delivered."[6]

As indicated in the Office action, Lambert teaches that "[a]ll queue managers support synchronous messaging operations, such as the ability to access queues to get, put and browse messages."[7] Lambert also discusses generating and sending a reply.[8]

The Office action fails to present a prima facie case of obviousness because the action does not teach or suggest many of the limitations of claims 1, 9, and 17. In particular, the combination of McGann and Lambert does not teach or suggest "storing the message in an inbound queue after receiving the message …." The Office action defines McGann's messaging collector 28 as the inbound queue. However, McGann describes the messaging collector as an interface rather than an inbound queue that stores the message until the queue is browsed and the message is copied. According to McGann, "[m]essaging collector 28 serves as an interface"[9] that "**immediately** passes the message off to local queue manager 30…."[10] The Office action does not address the lack of description of any functionality of the inbound queue in McGann but the Office action does assume functionality associated with the inbound queue. Applicant argues that an interface that "immediately passes the message off to local queue manager 30" does not teach or suggest an inbound queue as is suggested by the Office action to support the rejections.

---

[3] McGann, col. 2, lines 41-47.
[4] McGann, col. 2, lines 53-57.
[5] McGann, col. 2, lines 60-62.
[6] McGann, col. 3, lines 2-3.
[7] Lambert, par. 90.
[8] Lambert claims 1, 16, and 33; See also pg. 7, par 73; and pg 8, par. 96
[9] McGann col. 2, line 31.
[10] McGann, col. 2, lines 41-42.

Furthermore, when reading claim 1 in context, claim 4, which is dependent upon claim 1, describes "determining that the message is persisted prior to removing the message from the inbound queue". Again, the rejection of claim 4 simply assumes that determining the message is persisted prior to removing the message from the inbound queue is inherently taught by McGann and Lambert based upon the description in McGann of the messaging collector 28 as an "interface" that "**immediately** passes the message off to local queue manager 30…." Applicant argues that "determining that the message is persisted prior to removing the message from the inbound queue" is not taught or suggested by the discussions in McGann about the messaging collector 28, particularly when considering that the existence of the inbound queue is assumed.

The Office action argues that "[i]t is well known in the art that a message cannot be stored in a queue before it is received, thus it is understood that storing the message in an inbound queue after receiving the message is inherent." First, Applicant argues that the existence of the inbound queue in McGann's messaging collector 28 is not inherent in light of the description of the messaging collector. Second, storage in an inbound queue after receiving the message contradicts the description of an "interface" that "**immediately** passes the message off to local queue manager 30…." [11] In other words, McGann does not describe passing the message to the queue manager after storing the message in an inbound queue. McGann clearly states that the message immediately passes to local queue manager 30, which is more analogous to the function of "acceptor 132".[12] Thus, Applicant argues that the combination of McGann and Lambert fails to teach or suggest "storing the message in an inbound queue after receiving the message…."

The combination of McGann and Lambert fails to teach or suggest "browsing the inbound queue to identify the message after storing the message in the inbound queue…." The Office action relies on Lambert to provide the "browse" function. However, the combination of McGann and Lambert is improper because the combination requires a construction that contradicts a principle of operation explicitly described in McGann. In

[11] McGann, col. 2, lines 41-42.
[12] See Specification, pg. 6, par. 16, and Fig. 1.

this rejection, the Office action builds upon the assumption that messaging collector 28 comprises an inbound queue. Because Lambert teaches a browse function, the Office action argues that the combination of McGann and Lambert teaches a message collector 28 that receives the message, stores the message in an inbound queue after receipt, browses that inbound queue to identify the message prior to copying the message into a working queue, and then transmits the message to local queue manager 30. Applicant argues that this construction of McGann contradicts the description in McGann of the messaging collector 28 as an "interface" that "**immediately** passes the message off to local queue manager 30…."[13]

Furthermore, when reading claim 9 in context, claim 12, which is dependent upon claim 9, indicates that "the dispatcher is coupled with the working queue to wait until the thread is available to assign to the message or the thread is being cleaned, before copying the message to the working queue". Again, the rejection of claim 12 simply assumes that waiting until a thread is available to copy the message to the working queue is inherently taught by the combinations of McGann and Lambert based upon the description in McGann of the messaging collector 28 as an "interface" that "**immediately** passes the message off to local queue manager 30…"[14] and the "browse" function indicated by Lambert. Applicant argues that claim 9 is not taught or suggested by the combination.

The combination of McGann and Lambert fails to teach or suggest "copying the message to a working queue … to persist the message… stored, in it's entirety, in both the inbound queue and the working queue concurrently…." The Office action compounds the multiple assumptions above upon one another in the rejection of this element. First, the Office action assumes the existence of the inbound queue in messaging collector 28. Applicant notes that McGann does not describe a queue at all in the discussions of the messaging collector 28. Then, because McGann characterizes "**immediately** pass[ing] the message off to local queue manager 30…"[15] with "this happens very quickly", the Office action concludes that McGann is teaching or suggesting that the message is in both

[13] McGann, col. 2, lines 41-42.
[14] McGann, col. 2, lines 41-42.
[15] McGann, col. 2, lines 41-42.

an inbound queue and a working queue concurrently. Applicant argues that the speed with which actions occur offers no support for the conclusion that the message is in two queues concurrently.

The final Office action responds to this argument indicating that it is well known and inherent that a copy of a message is held in a [queue] at the delivery side of a system until the receiving end of the system receives. The examiner cites Maynard et al (US Pat. Pub. No. 2004/0153511) as support for this assertion. The problem with this assertion is that even if true, it's not applicable to the claim. The inbound, working, and outbound queues are part of a messaging system that receives the message from a requestor and processes the message. While the process may also involve transmitting messages, the inbound, working, and outbound queues are more analogous to the message queue 126 described in Maynard et al.[16] than they are to both queue 112 and queue 126. Modifying McGann to separate the messaging collector 28 from the local queue manager 30 by the communication network 106 described in Maynard changes a principle of operation of McGann and vice versa for Maynard et al. Note that the examiner relies on the proximity of the messaging collector 28 and local queue manager 30 to anticipate the "copying the message…" element of claims 1, 9, and 17.

The Background section alludes to the problem(s) addressed in the claims. In particular, paragraph 6 states that "persisting the message to non-volatile memory via upperware can… leave gaps in the persistence that may allow the message to be lost such as the gap between removing the message from the inbound queue and storing the message in non-volatile memory." The combination of the references cited in support of this rejection do not teach or suggest "copying the message to a working queue … to persist the message… stored, in it's entirety, in both the inbound queue and the working queue concurrently…" Applicant argues that the Office action fails to provide prima facie evidence for the rejections of claims 1, 9, and 17 so the rejections should be reversed.

---

[16] See Specification at Figs. 1-3, particularly server 130 and queues 134; and see Maynard et al. at Figs. 1 and 5, particularly queue 112 and queue 126.

The combination of McGann and Lambert does not teach or suggest "removing the message from the inbound queue after copying the message to the working queue". To support of the rejection, the Office action assumes the existence of an inbound queue and assumes that because McGann describes the message collector 28 as "**immediately** pass[ing] the message off to local queue manager 30..." [17], the message must be in both the inbound queue and a working queue concurrently. Based upon these assumptions, the Office action concludes that it is inherent that the message must be removed from the inbound queue after copying since the message was in the inbound queue concurrently with being in the working queue. Applicant argues that the Office action fails to provide prima facie evidence that the message is removed from the inbound queue after copying the message to the working queue, as is described in claims 1, 9, and 17.

And, the combination of McGann and Lambert does not teach or suggest "processing the message to generate a reply prior to removing the message from the working queue...." McGann describes queuing the message in a static internal class vector when the message is received by a thread of the local queue manager 30. McGann then states that "preferably, it is persisted to a local file system 32, where it is stored until the message is delivered. Until removed by the receiver, the message remains on the local file system so that it can be retrieved and resent in case of a hardware or software failure...." In essence, McGann states that the message remains in the local file system until the message is delivered.[18] McGann does not state that the message remains queued in the internal static vector until a reply is generated. And, McGann does not describe "processing the message ... prior to removing the message from the working queue...." McGann teaches that "[o]nce the message writer 36 has delivered the message, it removes the message from the local queue, where it had been placed by the local queue manager 30."[19] McGann does not teach or suggest "processing the message to generate a reply prior to removing the message from the working queue...."

---

[17] McGann, col. 2, lines 41-42.
[18] See also McGann at independent claims 1, 9, 17, and 21.
[19] McGann col. 3, lines 43-47.

Lambert discusses generating and sending a reply[20] as well as receiving a reply.[21] However, Lambert does not teach or suggest "processing the message to generate a reply prior to removing the message from the working queue…." Applicant argues that there is no basis in McGann, Lambert, or a combination thereof for the rejection so the Office action fails to provide a prima facie evidence of processing the message to generate a reply prior to removing the message from the working queue, as is described in claims 1, 9, and 17.

With respect to claims 2, 10, and 18, the combination of McGann and Lambert does not teach or suggest "removing the message from the working queue after storing the reply in an outbound queue." The Office action cites McGann at col. 3, line 43, which discusses the message but not a reply. While Lambert does discuss generation of a reply, Lambert does not teach or suggest this relationship between removing the message from the working queue and storing the reply in the outbound queue. Thus, Applicant argues that the Office action fails to provide prima facie evidence that the message is removed from the working queue after storing the reply in the outbound queue, as is described in claims 2, 10, and 18.

Furthermore, the dependents of claims 1, 9, and 17 incorporate the limitations of claims 1, 9, and 17. Thus, the combination of McGann and Lambert does not teach or suggest all the limitations of dependent claims of claims 1, 9, and 17, and Applicant respectfully argues that the dependent claims should be allowed.

The combination of McGann and Lambert fails to "teach or suggest all of Applicants' claim limitations" [22] so Applicant respectfully requests that the rejections of the claims be reversed.

---

[20] Lambert pg. 3, pars. 18 and 22; pg. 7, par. 73, pg. 8, par. 96; and claims 1, 5, 16, and 33.
[21] Lambert pg. 4, pars. 25, and 27-29; and claims 17, 25-26, and 28.
[22] *In re Royka*, 490 F.2d 981, 985, 180 USPQ 580, 583 (CCPA 1974).

Conclusion

To establish a prima facie case of obviousness, the modification or combination must teach or suggest all of Applicants' claim limitations.[23] The combination of McGann and Lambert does not teach or suggest all of Applicants' claim limitations. Thus, the rejections should be reversed as improper.

Authorization for an extension accompanies this brief. While no other fees are believed to be due, the Commissioner of Patents is hereby authorized to credit overpayments or debit underpayments via deposit account 09-0457.

Respectfully submitted,

April 20, 2009
Date

/Jeffrey S. Schubert/

Jeffrey S. Schubert, Reg. No. 43,098
Customer No.: 45557
Schubert Osterrieder & Nickelson PLLC
6013 Cannon Mtn. Dr, S14
Austin, Texas 78749
(512) 692-7297 (Telephone)
(512) 301-7301 (Facsimile)
Attorney for Appellant

[23] *In re Royka*, 490 F.2d 981, 985, 180 USPQ 580, 583 (CCPA 1974).

VIII. CLAIMS APPENDIX

TEXT OF CLAIMS PRESENTED ON APPEAL

WHAT IS CLAIMED IS:

1. A method for enhancing persistence of a message, the method comprising:
storing the message in an inbound queue after receiving the message;
browsing the inbound queue to identify the message after storing the message in the inbound queue;
copying the message to a working queue, the working queue being persisted by a queue manager, to persist the message, the message being stored, in it's entirety, in both the inbound queue and the working queue concurrently;
removing the message from the inbound queue after copying the message to the working queue;
processing the message to generate a reply prior to removing the message from the working queue; and
storing the reply in an outbound queue after generating the reply.

2. The method of claim 1, further comprising removing the message from the working queue after storing the reply in the outbound queue.

3. The method of claim 1, further comprising restoring the message in the working queue after a system failure.

4. The method of claim 1, further comprising determining that the message is persisted prior to removing the message from the inbound queue.

5. The method of claim 1, wherein browsing comprises searching the working queue for the message as part of a wave of messages in a chronologically adjacent order to facilitate generation of the reply, wherein the message is waiting to be processed.

6. The method of claim 1, wherein browsing comprises locking the message until the message is copied to the working queue.

7. The method of claim 1, wherein processing comprises assigning the message to a thread, the thread being available to process the message.

8. The method of claim 1, wherein processing comprises transmitting a second message to request data indicated by a content of the message and generating the reply based upon data received in response to the second message.

9. An apparatus for enhancing persistence of a message, the apparatus comprising:
an inbound queue to receive the message from a requestor;
a working queue to store the message;
an outbound queue to store a reply responsive to the message;
a queue manager to persist the message from the inbound queue and the working queue before the message is removed from the inbound queue; and
a dispatcher to browse the inbound queue to identify the message after the message is stored in the inbound queue; copy the message to the working queue to persist the message, the message to be stored, in it's entirety, in both the inbound queue and the working queue concurrently; remove the message from the inbound queue after the message is copied to the working queue and after the message is persisted from the working queue; and assign a thread to process the message to generate the reply in response to the message prior to removing the message from the working queue and to store the reply in the outbound queue after generating the reply.

10. The apparatus of claim 9, wherein the outbound queue is to store the reply until the reply is transmitted to the requestor.

11. The apparatus of claim 10, wherein the queue manager is configured to persist the message from the inbound queue and the reply from the outbound queue.

12. The apparatus of claim 9, wherein the dispatcher comprises a persistence determiner coupled with the queue manager to determine that the message is persisted prior to removing the message from the inbound queue, wherein the dispatcher is coupled with the working queue to wait until the thread is available to assign to the message or the thread is being cleaned, before copying the message to the working queue.

13. The apparatus of claim 9, wherein the dispatcher comprises a queue searcher to identify the message to be processed based upon priorities associated with messages stored in the inbound queue.

14. The apparatus of claim 9, wherein the dispatcher comprises a message locker to lock the message, until the message is copied into the working queue.

15. The apparatus of claim 9, wherein the dispatcher comprises recovery logic to assign the thread to process the message after a system failure.

16. The apparatus of claim 9, wherein the thread is configured to process the message based upon a rule associated with the message.

17. A storage-type machine-accessible medium containing instructions, which when executed by a machine, cause said machine to perform operations, comprising:

storing the message in an inbound queue after receiving the message;

browsing the inbound queue to identify the message after storing the message in the inbound queue;

copying the message to a working queue, the working queue being persisted by a queue manager, to persist the message, the message being stored, in it's entirety, in both the inbound queue and the working queue concurrently;

removing the message from the inbound queue after copying the message to the working queue;

processing the message to generate a reply prior to removing the message from the working queue; and

storing the reply in an outbound queue after generating the reply.

18. The storage-type machine-accessible medium of claim 17, wherein the operations further comprise removing the message from the working queue after storing the reply in an outbound queue.

19. The storage-type machine-accessible medium of claim 17, wherein the operations further comprise restoring the message in the working queue after a system failure based upon an order or priority associated with the message with respect to other messages in the working queue.

20. The storage-type machine-accessible medium of claim 17, wherein browsing comprises selecting a set of messages, the message being part of the set.

IX. EVIDENCE APPENDIX

Other than the Office Action(s), prior Appeal brief, and reply(ies) already of record, no additional evidence has been entered by Appellants or the Examiner in the above-identified application which is relevant to this appeal.

5

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings as described by 37 C.F.R. §41.37(c)(1)(x) known to Appellants, Appellants' legal representative, or assignee.